

## **AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently amended) A method executed in a data processing system for providing communication access between a first process associated with a first node and a second process associated with a second node, the method comprising:

sending a request from the first node to an administrative machine ~~a server-  
associated with a private network~~ to verify a first node identification  
associated with the first process;

in response to the request, receiving security context information at the first node  
from the administrative machine ~~server~~, the security context information  
comprising a virtual address for the first node;

appending the security context information for the first process in a process table;

opening a socket between the first process and the second process; and

transmitting a packet from the first process to the second process through the  
open socket without passing through the administrative machine ~~private-  
network~~, the packet comprising the security context information for the first  
process in the process table.

2. (Original) The method of claim 1, further comprising modifying a socket structure so as to accept the security context information.

3. (Original) The method of claim 1, further comprising:

receiving the packet at the second process through the socket;

verifying the security context information received in the packet; and

permitting use of the packet if the security context information is verified.

4. (Canceled)

5. (Previously presented) The method of claim 46, wherein determining if the first and second process belong to a channel comprises:

comparing the security context information in the received packet and security context information in another process table.

6. (Canceled)

7. (Previously presented) The method of claim 3, wherein verifying the security context information comprises:

determining whether the first and second process belong to two different linked channels; and

permitting use of the packet when the different channels are linked.

8. (Previously presented) The method of claim 7, wherein determining whether the first and second process belong to two different linked channels comprises:

initiating a process that spawns two child processes that are connected by a shared-memory region in a memory.

9. (Previously presented) The method of claim 7, wherein permitting use of the packet comprises:

decrypting the packet; and

authenticating a sender associated with the first process.

10. (Previously presented) The method of claim 1, wherein appending security context information comprises:

obtaining the security context information from a third process, the security context information comprising a virtual address and a node identification.

11. (Original) The method of claim 1, further comprising:

modifying a network stack such that the network stack requires the security context information to be present in the socket structure to transmit.

12. (Canceled)

13. (Previously presented) The method of claim 1, wherein receiving security context information further comprises:

receiving a key that corresponds to the first node identification from the server.

14. (Previously presented) The method of claim 13, further comprising:

encrypting a packet transmitted by the first process using the key;

encapsulating the encrypted packet with a header that comprises the first node identification.

15. (Previously presented) The method of claim 1, further comprising:

sending a second request from the second node to the server to verify a second node identification;

receiving additional security context information from the server, wherein the additional security context information comprises a second virtual address for the second node;

creating the second process; and

appending the security context information for the second process in the process table associated with the second process.

16. (Currently amended) A method executed in a data processing system for providing secure communications between a first process associated with a first node and a second process associated with a second node, the method comprising:

obtaining a node identification comprising a virtual address from an

administrative machine ~~a server associated with a private network~~;

including the node identification in a field corresponding to the first process in a process table;

transmitting a datagram that contains the node identification from the first process to a socket; and

receiving the datagram at the second process that contains the node identification and a second virtual address, without the datagram passing through the administrative machine ~~private network~~.

17. (Previously presented) The method of claim 16, wherein obtaining a node identification further comprises:

modifying a socket structure in the socket so that the socket structure accepts the node identification; and

modifying a process table so that the table comprises a node identification field.

18. (Currently amended) A system for providing communication access between a first process associated with a first node and a second process associated with a second node, comprising:

means for sending a request from the first node to an administrative machine ~~a server associated with a private network~~ to verify a first node identification associated with the first process;

means for receiving security context information, in response to the request, at the first node from the administrative machine server, the security context information comprising a virtual address for the first node;

means for appending the security context information for the first process in a process table;

means for opening a socket between the first process and the second process;

and

means for transmitting a packet from the first process to the second process through the open socket without passing through the administrative machine private network, the packet comprising the security context information for the first process in the process table.

19. (Original) The system of claim 18, further comprising means for modifying a socket structure so as to accept the security context information.

20. (Original) The system of claim 18, further comprising:

means for receiving the packet at the second process through the socket;

means for verifying the security context information received in the packet; and

means for permitting use of the packet if the security context information is verified.

21. (Canceled)

22. (Previously presented) The system of claim 47, wherein means for determining if the first and second process belong to a channel comprises:

means for comparing the security context information in the received packet and security context information in another process table.

23. (Canceled)
24. (Previously presented) The system of claim 20, wherein means for verifying the security context information comprises:
- means for determining whether the first and second process belong to two different linked channels; and
  - means for permitting use of the packet when the different channels are linked.
25. (Previously presented) The system of claim 24, wherein means for determining whether the first and second process belong to two different linked channels comprises:
- means for initiating a process that spawns two child processes that are connected by a shared-memory region in a memory.
26. (Previously presented) The system of claim 24, wherein means for permitting use of the packet comprises:
- means for decrypting the packet; and
  - means for authenticating a sender associated with the first process.
27. (Previously presented) The system of claim 18, wherein means for appending security context information comprises:
- means for obtaining the security context information from a third process, the security context information comprising a virtual address and a node identification.
28. (Original) The system of claim 18, further comprising:
- means for modifying a network stack such that the network stack requires the security context information to be present in the socket structure to transmit.

29. (Currently amended) A system for placing a process executed in a node in a security context, comprising:

an administrative machine ~~a server associated with a private network~~; and

a sending node comprising:

a transmission module that transmits a request to the administrative

machine ~~server~~ to verify a sending node identification, and receives

security context information from the administrative machine ~~server~~

in response to the request, wherein the security context information

comprises a virtual address for the sending node;

memory containing a process and an associated process table; and

an appending module that appends the received security context

information and the sending node identification for the process in

the process table, wherein the transmission module transmits a

packet from the process to a receiving node without passing

through the administrative machine ~~private network~~, the packet

comprising the security context information for the first process in

the process table.

30. (Previously presented) The system of claim 29, wherein the transmission module further receives a key that corresponds to the sending node identification from the administrative machine ~~server~~.

31. (Previously presented) The system of claim 30, further comprising:

an encryption module that encrypts the packet transmitted by the process using

the key; and

an encapsulating module that encapsulates the encrypted packet with a header that comprises the sending node identification.

32. (Canceled)

33. (Currently amended) A system for providing secure communications between a first process associated with a first node and a second process associated with a second node, comprising:

means for obtaining a node identification comprising a virtual address from an administrative machine ~~a server associated with a private network~~;

means for including the node identification in a field corresponding to the first process in a process table;

means for transmitting a datagram that contains the node identification from the first process to a socket; and

means for receiving the datagram at the second process that contains the node identification and a second virtual address, without the datagram passing through the administrative machine ~~private network~~.

34. (Previously presented) The system of claim 33, wherein means for obtaining a node identification further comprises:

means for modifying a socket structure in the socket so that the socket structure accepts the node identification; and

means for modifying a process table so that the table comprises a node identification field.

35. (Currently amended) A computer readable medium for controlling a data processing system to perform a method for providing communication access between a



first process associated with a first node and a second process associated with a second node, comprising:

- a sending module for sending a request from the first node to an administrative machine ~~a server associated with a private network~~ to verify a first node identification associated with the first process;
- a receiving module for receiving security context information, in response to the request, at the first node from the administrative machine ~~server~~, the security context information comprising a virtual address for the first node;
- an appending module for appending security context information for the first process in a process table;
- an opening module for opening a socket between the first process and the second process;
- a transmitting module for transmitting a packet from the first process to the second process through the open socket without passing through the administrative machine ~~private network~~, the packet comprising the security context information for the first process in the process table.

36. (Original) The computer readable medium of claim 35, further comprising a modifying module for modifying a socket structure so as to accept the security context information.

37. (Original) The computer readable medium of claim 35, further comprising:  
a receiving module for receiving the packet at the second process through the socket;

a verifying module for verifying the security context information received in the packet; and

a permitting module for permitting use of the packet if the security context information is verified.

38. (Canceled)

39. (Previously presented) The computer readable medium of claim 48, wherein the determining module comprises:

a comparing module that compares the security context information in the received packet and security context information in another process table.

40. (Canceled)

41. (Previously presented) The computer readable medium of claim 37, wherein the verifying module comprises:

a determining module for determining whether the first and second process belong to two different linked channels; and

a permitting module for permitting use of the packet when the different channels are linked.

42. (Previously presented) The computer readable medium of claim 41, wherein the determining module comprises a initiating module that initiates a process that spawns two child processes that are connected by a shared-memory region in a memory.

43. (Previously presented) The computer readable medium of claim 41, wherein the permitting module comprises:

a decrypting module for decrypting the packet; and

an authenticating module for authenticating a sender associated with the first process.

44. (Previously presented) The computer readable medium of claim 35, wherein the appending module comprises:

an obtaining module for obtaining the security context information from a third process, the security context information comprising a virtual address and a node identification; and

a limiting module for limiting each of the first, second and third processes to communicate with another process provided that the communicating processes share the same node identification.

45. (Original) The computer readable medium of claim 35, further comprising:

a modifying module for modifying a network stack such that the network stack requires the security context information to be present in the socket structure to transmit.

46. (Previously presented) The method of claim 1, further comprising

determining if the first and second process belong to a channel; and

accepting the transmitted packet when the first and second process belong to the channel.

47. (Previously presented) The method of claim 18, further comprising

means for determining if the first and second process belong to a channel; and

means for accepting the transmitted packet when the first and second process belong to the channel.

48. (Previously presented) The computer-readable medium of claim 35, further comprising:

a determining module for determining if the first and second process belong to a channel; and  
an accepting module for accepting the transmitted packet when the first and second process belong to the channel.